Description

REAL–TIME REMOTE BACKUP SYSTEM AND RELATED METHOD

BACKGROUND OF INVENTION

[0001]   1. Field of the Invention

[0002]   The present invention relates to a real–time remote backup system and a related method, and more specifically, to a real–time network data remote backup system and a related method which are capable of dynamically backing up file modification among a plurality of LINUX computer devices.

[0003]   2. Description of the Prior Art

[0004]   As network technology progresses, Internet appliances (or called " information appliances"), such as a network attached storage (NAS) device composed of redundant arrays of inexpensive disks (RAID), are newly developed. Such NAS devices can be connected directly to a network system for allowing a remote user's computer system

(source terminal) to store or download a large amount of data to/from the network storage device. In addition to user computer data storage, the NAS device can also be used as a remote backup system. However, in order to provide independent remote backup, the NAS device does not only need a random access memory (RAM) or a flash ROM, a plurality of hard disk drives (HDD) and IC controllers but also needs a specific embedded operating system (OS) and related applications. However, these OS and applications are all designed variously on base of demands of different manufacturers.

[0005] An automatic network backup software, called "RSYNC", has been introduced. The software was first written by Andrew Tridgell, and successively modified by other programmers all around the world. This software is a freeware under general public license of GNU, which is easy to install and able to perform rapid backups. A computer system having RSYNC installed can be a client or a server, or both at the same time. When the backup process is executed on two computer systems having RSYNC installed, such as a client and a server, a checksum test will be performed according to an RSYNC algorithm, and then the client can setup a backup time to check the checksum in

order to find modified parts of files and perform a backup to the server instead of transmitting the whole file. In such a manner, the transmission speed and the efficiency of the file backup are improved.

[0006] However, in order to improve the backup ability of RSYNC, the following items are suggested: Rather than performing periodic checks for file modifications, if any file modification event occurs, a client computer system in a Linux environment should immediately notify RSYNC to backup the modified file. However, if it is required to dynamically catch any file modification message and automatically notify RSYNC to perform a backup, in a Linux environment, the kernel of the OS is also required to be rewritten and debugged. This is a very complicated task for a programmer.

SUMMARY OF INVENTION

[0007] It is therefore a primary objective of the present invention to provide a real-time remote backup system and a related method utilized among a plurality of computer systems in a Linux environment. The system and the method can dynamically detect file modification events in a user space of a source computer system and automatically notify network backup software such as RSYNC to backup

only the modified part to a destination computer system. Therefore, it can reach the objects that there are the reduced data loading and the faster transmission speed to raise the backup efficiency. Moreover, the system and the method can approach the synchronal update of the files between the source computer system and a destination computer system, as well as notify the programmer of the file modification without the need of rewriting the kernel of Linux.

[0008] Briefly summarized, a real-time remote backup system used in a network system interconnecting between at least one source computer system and one destination computer system is disclosed. Each computer system consists of a kernel space and a user space. The real-time remote backup system includes a loadable kernel module that pre-sets setting up at least one specific system call in the kernel space of the source computer system. The real-time remote backup system call notifies the loadable kernel module to generate a corresponding file modification message when a file modification event occurs in the user space of the source computer system. A scheduling module is used for sequentially queuing the file modification message from the loadable kernel module and then gen-

erating a corresponding backup command in response to the file modification message. At least one network backup unit is installed in the source computer system for backing-up the modified part of the file through the network system to the destination computer system when receiving the backup command from the scheduling module.

[0009] The present invention further provides a real-time remote backup system installed in a source computer system connected to a destination computer system through a network system. Each computer system includes at least one kernel space for a kernel and a user space for a network backup unit. The real-time remote backup system includes a loadable kernel module that pre-sets up at least one specific system call in the kernel space of the source computer system with removal of an original system call. The loadable kernel module generates a file modification message according to the type of the specific system call when a file modification event occurs to facilitate the specific system call notify the kernel. A scheduling module is for queuing the file modification message from the loadable kernel module and processing each file modification message to generate a corresponding

backup command to the network backup unit of the source computer system so that the network backup unit backs-up the modified file to the destination computer system through the network system.

[0010] The present invention further provides a method of real-time remote backup used in a network system interconnecting between at least one source computer system and one destination computer system. Each computer system consists of a kernel space and a user space. The real-time remote backup method includes: implementing a specific system call that is pre-determined by a loadable kernel module loaded in the kernel space, to notify a kernel of the source computer system to modify a file; generating a file modification message according to the type of the specific system call which also notifies the loadable kernel module of a file modification event; queuing in sequence the file modification messages into a queue unit; sequentially taking the file modification messages from the queue unit to process and generate a corresponding backup command; and backing-up the modified part of the file by the network backup unit to the destination computer system through the network according to the backup command.

[0011] These and other objectives of the present invention will no doubt become obvious to those of ordinary skill in the art after reading the following detailed description of the preferred embodiment that is illustrated in the various figures and drawings.

BRIEF DESCRIPTION OF DRAWINGS

[0012] Fig.1 illustrates a main structure of a real-time remote backup system utilized in a network system according to the present invention.

[0013] Fig.2 illustrates a detailed structure of the real-time remote backup system shown in Fig.1.

[0014] Fig.3 is a flowchart of a method of real-time remote backup according to the present invention.

DETAILED DESCRIPTION

[0015] Please refer to Fig.1 showing a real-time remote backup system 100 utilized with a network system 20. The network system 20 interconnects between at least one source computer system 10 and a destination computer system 30 in a Linux environment 30. Each computer system 10, 30 consist of a kernel space 102 having a kernel 108, and a user space 104.

[0016] The real-time remote backup system 100 has at least one

loadable kernel module 110 installed in the kernel space 102 of the source computer system 10 for detecting whether a file modification event 106 has occurred in the user space 104. A scheduling module 120 installed in the user space 104 of the source computer system 10 sequentially queues each file modification message transmitted from the loadable kernel module 110 and process the file modification message to generate a backup command. A pair of network backup units 130, 302, such as a pair of RSYNC applications, is respectively installed in the user spaces 104 of the source computer system 10 and the destination computer system 30 to allocate the computer systems to be as a client and a server. When receiving the backup command from the scheduling module 120, the network backup unit 130 provides file modification backups to the network backup unit 302 of the destination computer system 30.

[0017] As shown in Fig.2, the loadable kernel module 110 includes a replacement unit 112, a call determining unit 114 and a message processing unit 116. The replacement unit 112 replaces specific original system calls in the kernel space 102 with new system call 109 defined by the user. Actually, each system call is implemented as a function

stored permanently in a memory. Whenever any file modification event occurs, such as deleting/saving a file, deleting/establishing a directory, or closing a file, a corresponding system call has to be implemented to notify the kernel 108 of the file variance, as known. By way of replacing with the new system call 109, the notification of file modification event therefore can be fast obtained without need of modifying the kernel 108. When the new system call 109 is implemented, the loadable kernel module 110 can also be notified immediately of the file modification event. In addition, the real-time remote backup system 100 can include a graphical user interface (GUI, not shown) disposed with a network automatic backup switch for providing the user to switch on/off an automatic network backup function. When the user switches off the automatic network backup function, the replacement unit 12 of the loadable kernel module 100 will switch back to the original system call.

[0018] As described above, when the new system call 109 notifies the loadable kernel module 100, the call determining unit 144 of the loadable kernel module 100 first determines whether the new system call 109 is one of the predetermined system calls, such as SYS_RMDIR, SYS_MKDIR,

SYS_RENAME, SYS_LINK, SYS_CHMOD, etc. If yes, the message processing unit 116 generates a corresponding file modification message including at least the filename and path of the modified file, to the scheduling module 120. If no, the system call is ignored and the process is terminated.

[0019]   The scheduling module 120 includes a schedule managing unit 122, a queue unit 124 and a schedule processing unit 126. The schedule processing unit 14 and the schedule managing unit 126 use the same scheduling algorithm to process said file modification message generated by the message processing unit 116 of the loadable kernel module 110. The scheduling algorithm must be coordinated with the algorithm of the network backup unit 130 in order to prevent data from losing during data processing/transmission per step. Using the scheduling algorithm, the schedule managing unit 122 queues the received messages in sequence into the queue unit 124, and the schedule processing unit 126 reads out and processes each file modification message from the queue unit 124 in order to generate a corresponding backup command to the network backup unit 130. The backup command includes at least the path of the modified files. Additionally,

the schedule processing unit 126 and the schedule managing unit 122 respectively have specific threads at least one of which is used to define an adjustable optimized time interval to limit and coordinate the message input/output and command transmission of the two units 122, 126 in the queue unit 124. However, as is known by a person skilled in the art, a device combining the two units 122, 126 would also belong to the score of the present invention.

[0020] When receiving the command corresponding to the file modification from the scheduling module 120, the network backup unit 130 executes a checksum test in the source computer system 10 according to the path of the modified file provided by the command in order to find the difference between the two files in the source computer system 10 and the destination computer system 30. Only the modified part is then transmitted to the destination computer system 30 through the network system 20.

[0021] Please refer to Fig.3 showing a method of real-time remote backup according to the present invention, including the following steps:

[0022] In step 60, replacing at least one original system call in the kernel with the new system call 109 in the loadable

kernel module 110.

[0023] In step 61, a file modification event occurs, such as deleting/saving a file, deleting/establishing a directory, or closing a file, in the user space 104 of the source computer system 10. In step 62, implementing a corresponding predetermined system call 109 to notify the kernel 108 in the kernel space 102 for proceeding in Step 64, i.e. modify the file.

[0024] In step 65, when the new system call 109 is implemented, also notifying the call determining unit 114 of the loadable kernel module 110 of the file modification event.

[0025] In step 70, the call determining unit 114 of the loadable kernel module 110 determines whether the new system call 109 is one of the predetermined system calls, such as SYS_RMDIR, SYS_MKDIR, SYS_RENAME, SYS_LINK, SYS_CHMOD, etc. If yes, proceed to Step 72, the message processing unit 116 generates and sends out a file modification message including at least the filename and the path of the modified file to the scheduling module 120. And if no, proceed to Step 74, i.e. end.

[0026] In step 76, the scheduling module 120 sequentially queues the received messages into the queue unit 124.

[0027] In step 80, the scheduling module 120 reads/takes the

file modification messages in sequence from the queue unit 124 to generate the backup command to the network backup unit 130. The command includes at least the path of the modified file.

[0028] In step 85, when receiving the command related to file modification from the scheduling module 120, the network backup unit 130 executes the checksum test in the source computer system 10 according to the path of the modified file provided by the command in order to find the difference between the two files in the source computer system 10 and the destination computer system 30. It then transmits only the modified part to the destination computer system 30 through the network system 20. However, the real-time remote backup system 100 according to the present invention is not limited to a network having a server and a client. A network having a plurality of servers and clients also belongs to the present invention.

[0029] In contrast to the prior art, the real-time remote backup system 100 and the related method according to the present invention dynamically captures file modification events to transmit them to the network backup unit 130. This is in contrast to, as the prior art, performing backups

after a specific time period or searching the whole source computer system 10 to fine modified files in order to backup at once. Therefore, the efficiency of the backup and the transmission speed of the network are improved.

[0030] Those skilled in the art will readily observe that numerous modifications and alterations of the device may be made while retaining the teachings of the invention. Accordingly, the above disclosure should be construed as limited only by the metes and bounds of the appended claims.